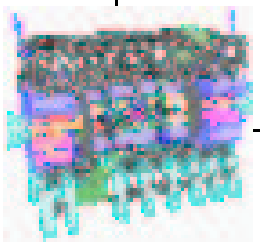
A User's Experience with the Testbed Software

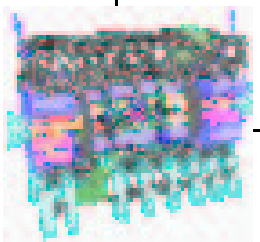
Outline:

- Status of OUHEP Testbed Site
- Testbed Software Installation
- Suggested Installation Improvements
- Running the User Software
- Suggested Running Improvements
- Summary
- Bonus: OU Network Performance

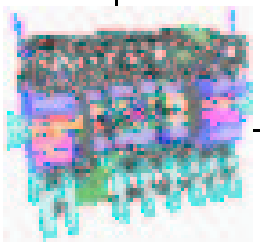


Status of OUHEP Testbed Site, ouhep0.nhn.ou.edu

- 9 Node, 15 Processor Cluster: 1 Dual Athlon 1.5 GHz Processor (2 GB RAM), 5 Dual P-III (1 GHz and 550 MHz) Processor (1 GB RAM), 3 Single P-II (233 MHz) Processor (64 MB RAM) Machines
- 1 TB of disk space (800 GB RAID array + 200 GB SCRATCH and ATLAS software)
- 1000 Mbps connection into RAID server / gatekeeper
- Running RedHat 7.2, 2.4.9-34 Kernel
- All together 2.5 GFlops



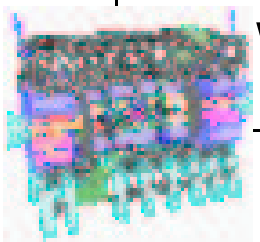
- OpenAFS 1.2.5, Globus 2.0b, gsi-wuftpd 0.5, Condor 6.3.1, iperf 1.2
- MDS: GIIS Server on gatekeeper, reporting on system status and Condor jobmanager
- ATLAS software 'installed' via AFS, Atlfast (3.0.1) locally
- Testbed V.2 installed with Pacman
- Grid User accounts policy: group
- System Security: only ssh entry allowed
- Shared with D0



Testbed Software Installation

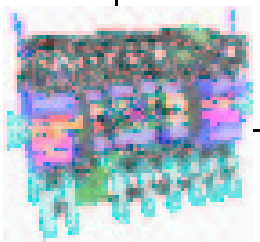
Three Packages installed:

- **VDTgrid** only installed new **Pacman** and **pippy**, and updated Pacman db with **Globus** and **atlfast** location
- **Atlas** installed **Magda**, **Atlsim**, **Grappa boxed input**, and **java**, **perl**, **mysql**
- **Atlas-user** installed **Grappa** and **Commandline Toolkit**
- Packages installed fine on most sites –except for some setup issues with hard wired variables – except OU, where we had a conflict with already installed D0 (newer) version of perl
- Therefore I created and installed **-noperl-nojava** version, since we already had standard RH perl/mysql and jdk rpm installed

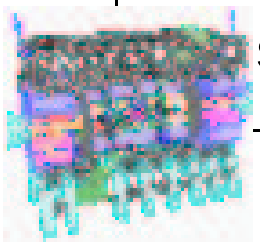


Suggested Installation Improvements

- Ideally, only one copy of Pacman, so that a Pacman update won't have to be done multiple times
- Need **perl-native**, **mysql-native**, and **jdk-native** packages; use installed native software **WHEREVER POSSIBLE**; multiple installations of standard software a **REALLY BAD IDEA!** (See problems with DC1)
- Also, do not attempt to overwrite system software during pacman installs as root
- Check PATH before adding anything; no use having something twice in the path

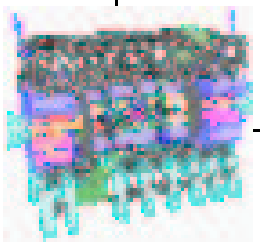


- Double check that all env vars and **PATHs** are actually necessary (e.g., **$PERL_BUILD** and **$MYSQL_BUILD** areas seem to be empty)
- No hard wiring of native locations (i.e, **$GLOBUS_LOCATION**) in setup scripts
- Make sure to keep sh and csh syntax straight – right now, offenders both Pacman (VDTgrid: GLOBUS_LOCATION) and Grappa (Atlas-user (jakarta-ant-1.4.1-nojava): JAVACMD)
- If possible, don't completely overwrite **setup.*sh** scripts with every pacman operation, because users may have added customizations or commented out unnecessary parts
- Improvement of -remove(all), which doesn't seem to remove software (already in newer version, I hear)

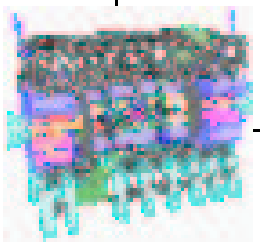


Running the Toolkit Software

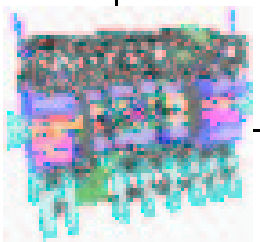
- Toolkit very easy to use:
- **./run-atlfast-production all 10000 all**
- will produce 10000 events each of all 5 Physics processes on all sites registered (currently BNL, IUPUI, LBL, OU, UTA)
- $\Longrightarrow$ 250000 events, stored at MAGDA caches
- best output location discovered automatically
- problems only with stability of some sites (network, lsf, overload, ...)



- Grappa needs a lot more steps to get started:
  - **portal-start**
  - connect to https server, login
  - go to Notebook Menu
  - select or import Notebook
  - importing didn't work for me, I got three different errors
  - list sessions
  - open Athena Notebook
  - not clear how to view or choose Physics processes, number of events
  - submit – enter Input Location, basename, start and end, Output Location, select resource, submit, sub-notebook name, submit

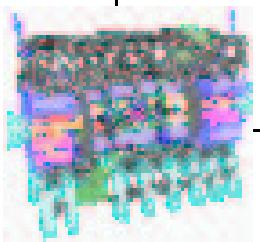


- Finally, it runs!
- not clear how to look at plots, though
- Also, looks like it's still copying boxed input from remote site, rather than using local (/opt/atlfast/) copy

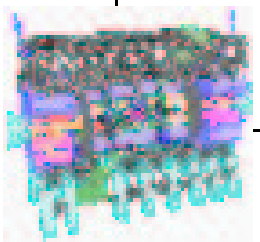


Suggested Running Improvements

- Commandline Toolkit:
  - More documentation
  - Better logging of production data back to submitter
  - Metadata
  - Better error logging and handling and cleanup
  - More stability needed at some sites (BNL, IU)
  - Adding all testbed sites (right now only 5)
  - Distributed Analysis Tools

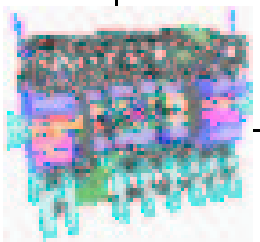


- Grappa:
  - Better step-by-step documentation on how to run it
  - Production notebook in addition to user notebook
  - Automatic file naming
  - Automatic output file location discovery
  - If possible, java required only on gatekeeper, not on every compute node

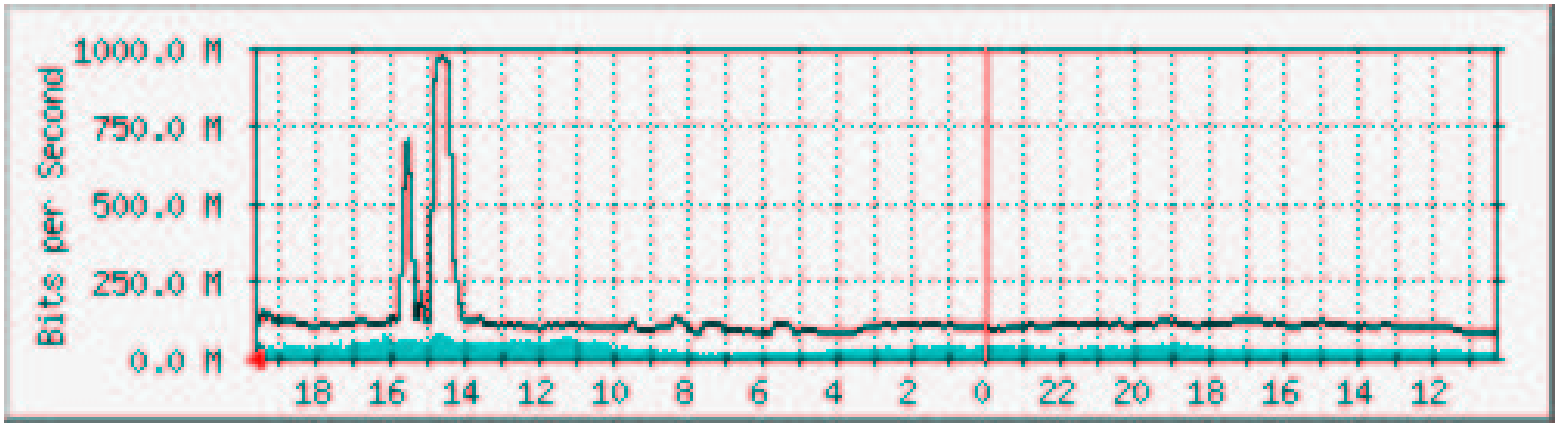


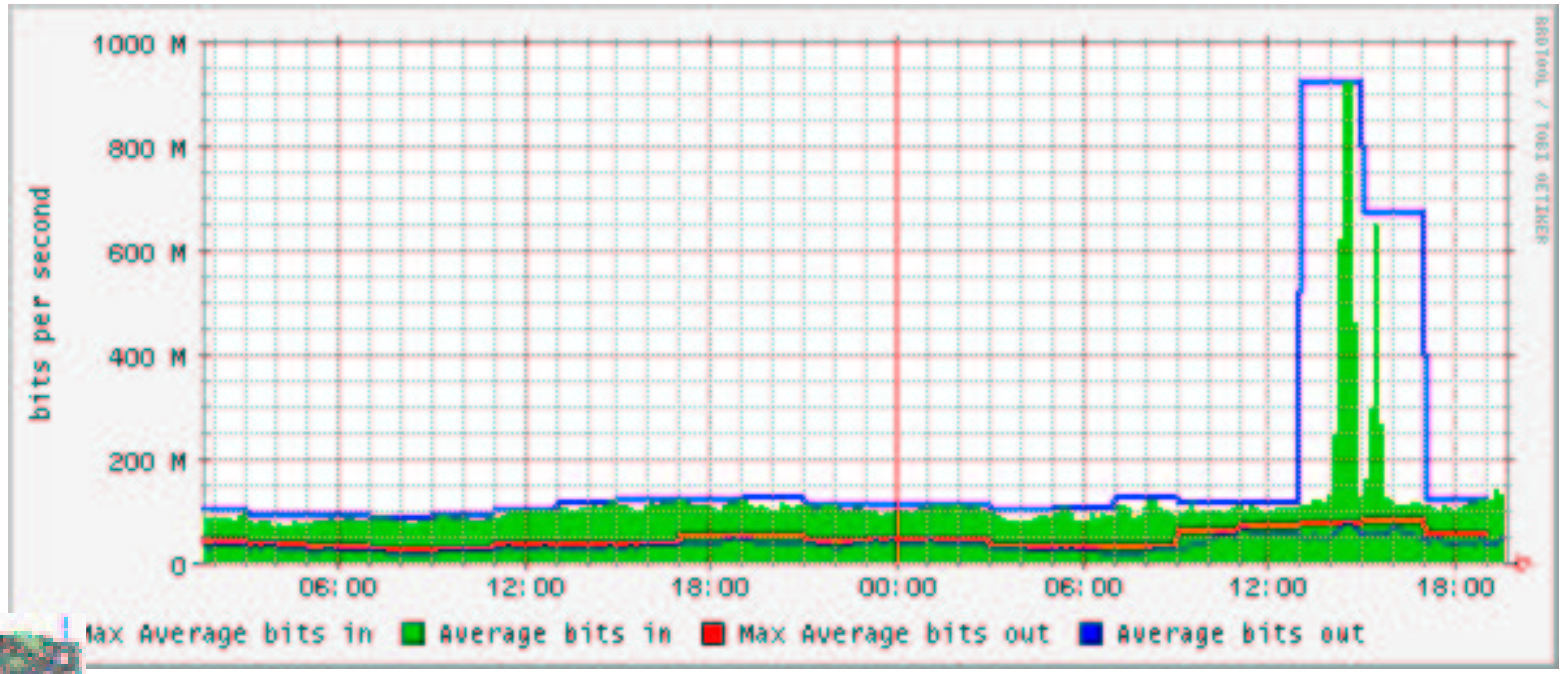
Summary

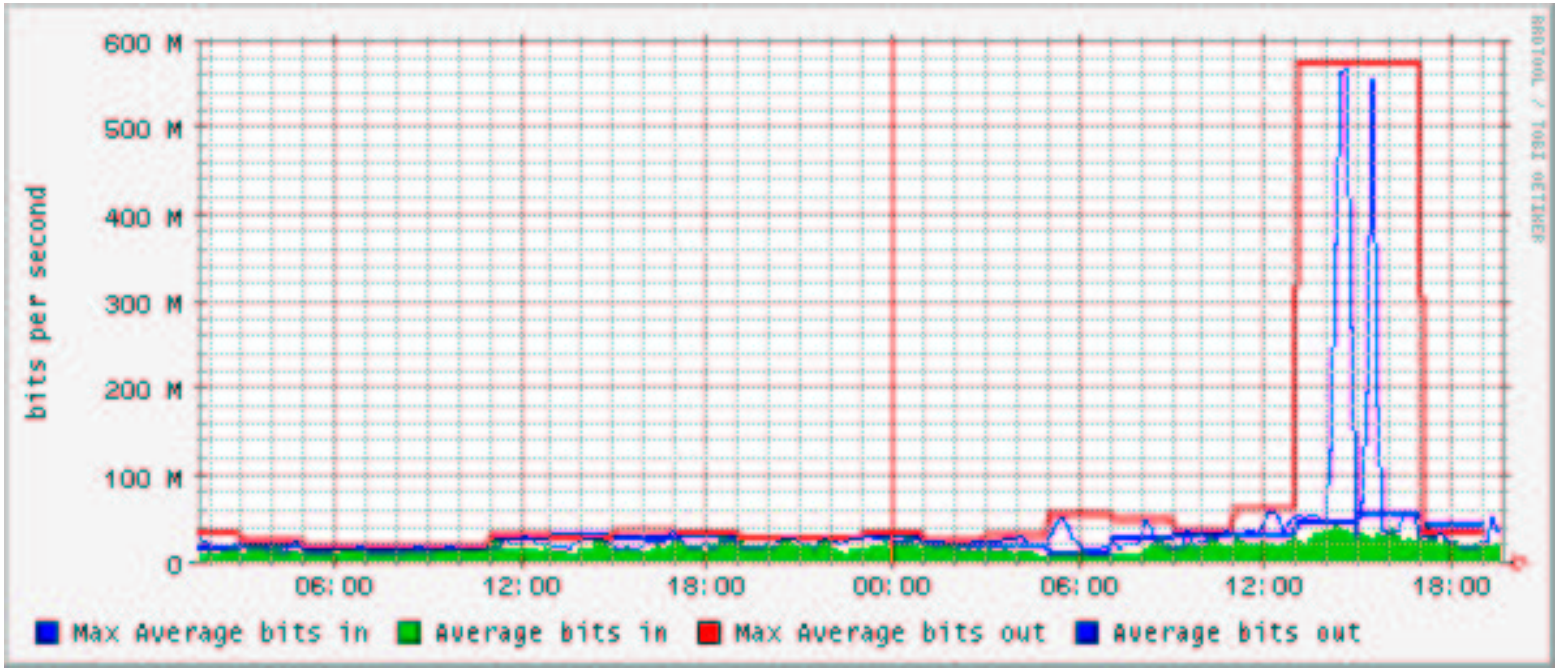
- Installations went pretty painless, just minor snags
- Software performs quite well, but needs:
  - Additional features
  - More logging/debugging information
  - More robustness
  - Better documentation
- Thanks to Saul for a great deal of work with Pacman
- Thanks to Dan and Shava for great Grappa development
- Thanks to Kaushik for Toolkit development and lots of helpful hints



OU Network Performance






600 M
500 M
400 M
300 M
200 M
100 M
0
bits per second
06:00
12:00
18:00
00:00
06:00
12:00
18:00
Max Average bits in
Average bits in
Max Average bits out
Average bits out